PROGRAMAÇÃO PARALELA EM MACHINE LEARNING

Igor Freitas
Intel

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.

Performance results are based on testing as of Aug. 20, 2017 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Any forecasts of goods and services needed for Intel's operations are provided for discussion purposes only. Intel will have no liability to make any purchase in connection with forecasts published in this document.

ARDUINO 101 and the ARDUINO infinity logo are trademarks or registered trademarks of Arduino, LLC.

Altera, Arria, the Arria logo, Intel, the Intel logo, Intel Atom, Intel Core, Intel Nervana, Intel Saffron, Iris, Movidius, OpenVINO, Stratix and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Programação Paralela em IA/ML Vs Programação Paralela “tradicional” (HPC)

Programação Paralela em ML Frameworks

Exemplo de código

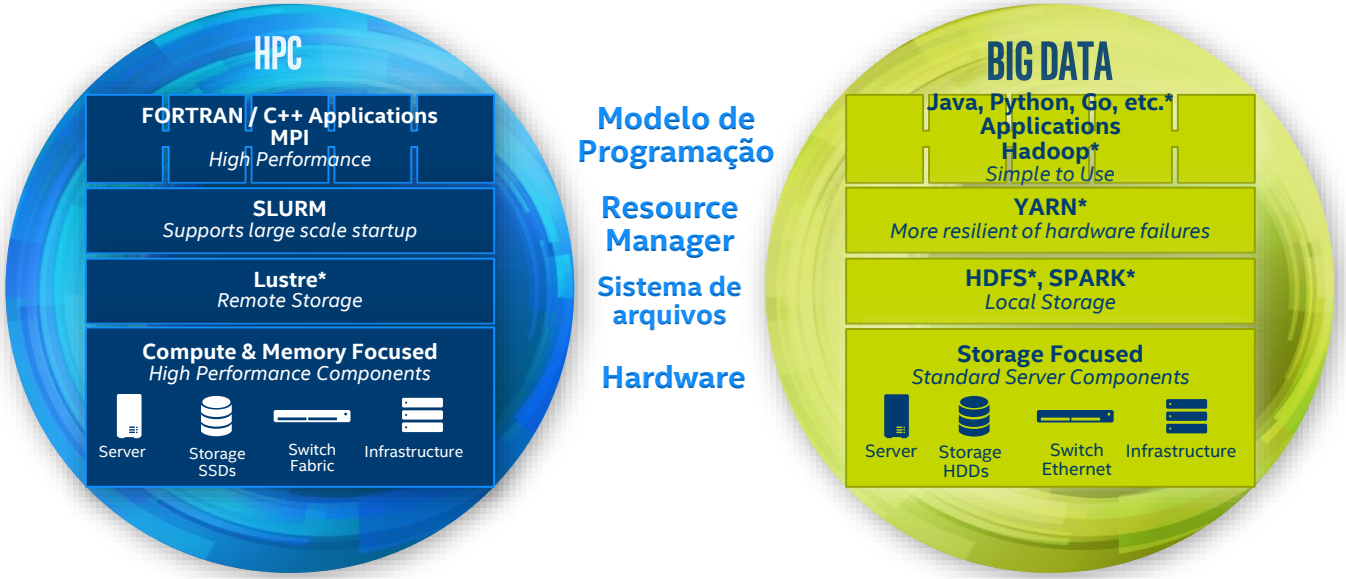
Programação Paralela em IA/ML Vs Programação Paralela “tradicional” (HPC)

Programação Paralela em ML Frameworks

Exemplo de código

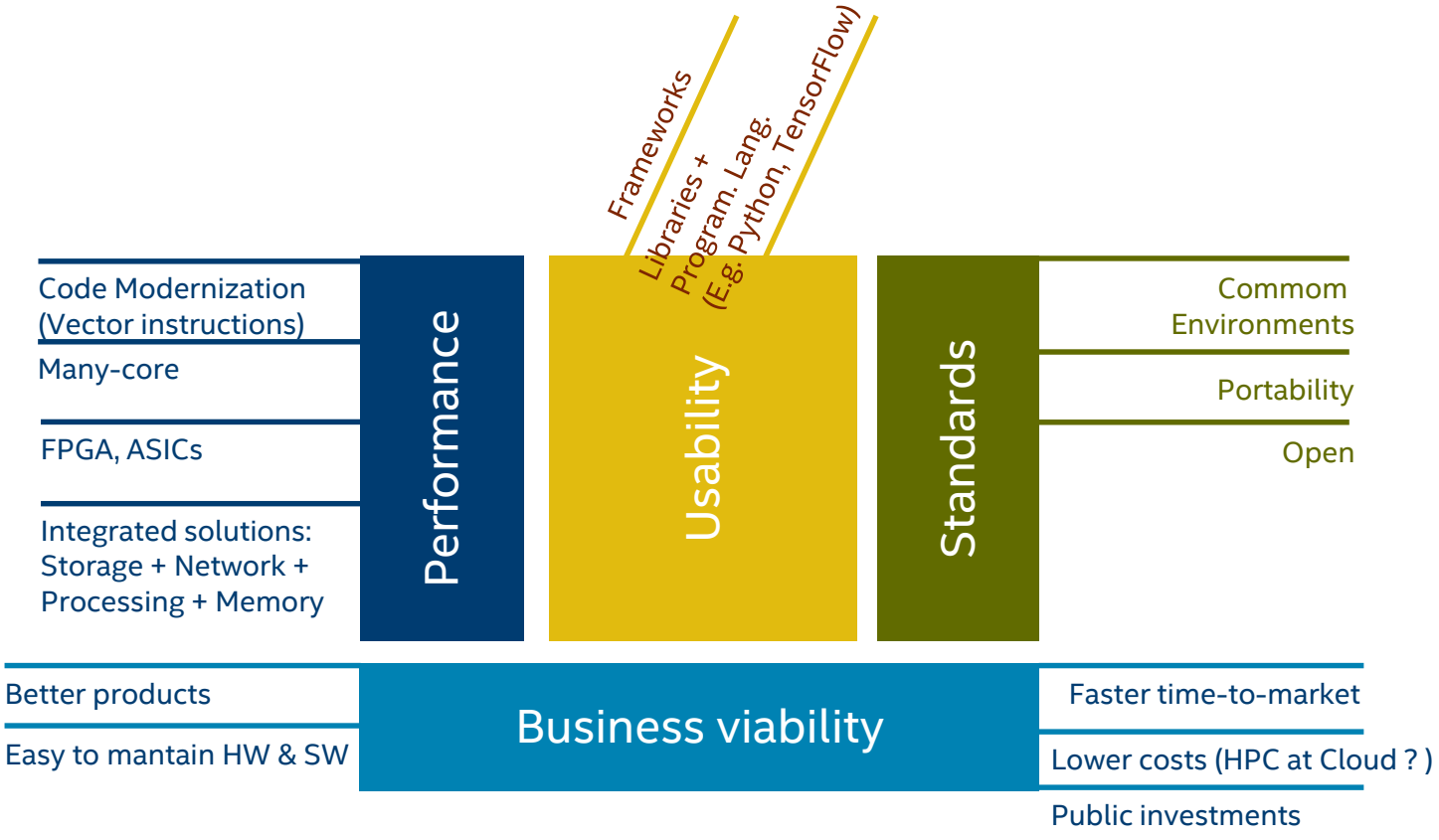
Big Data Analytics

HPC != Big Data Analytics != Inteligência Artificial ?



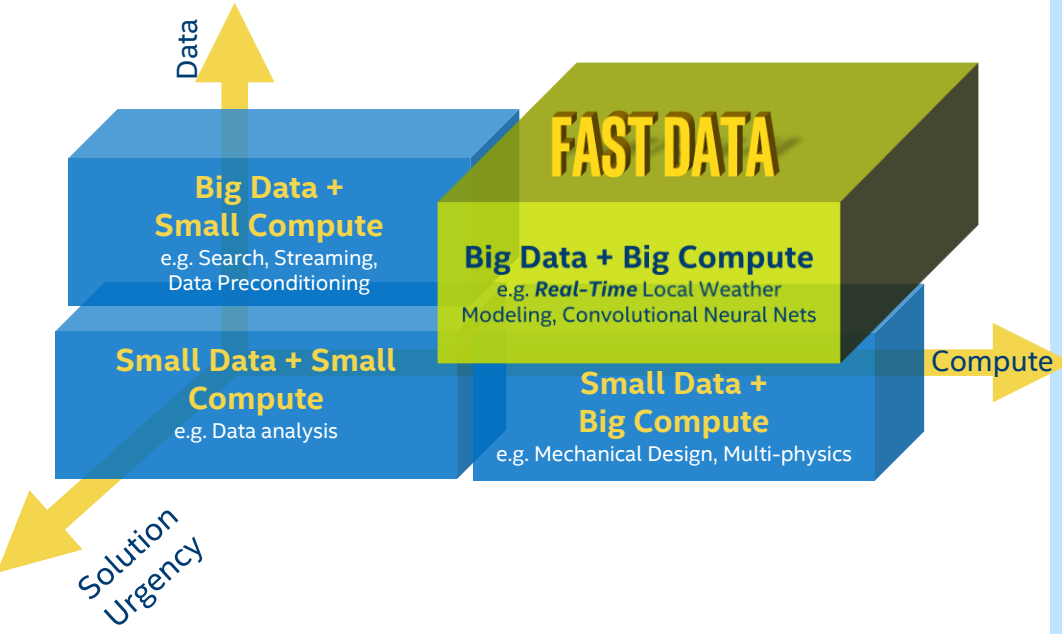
*Other brands and names are the property of their respective owners.

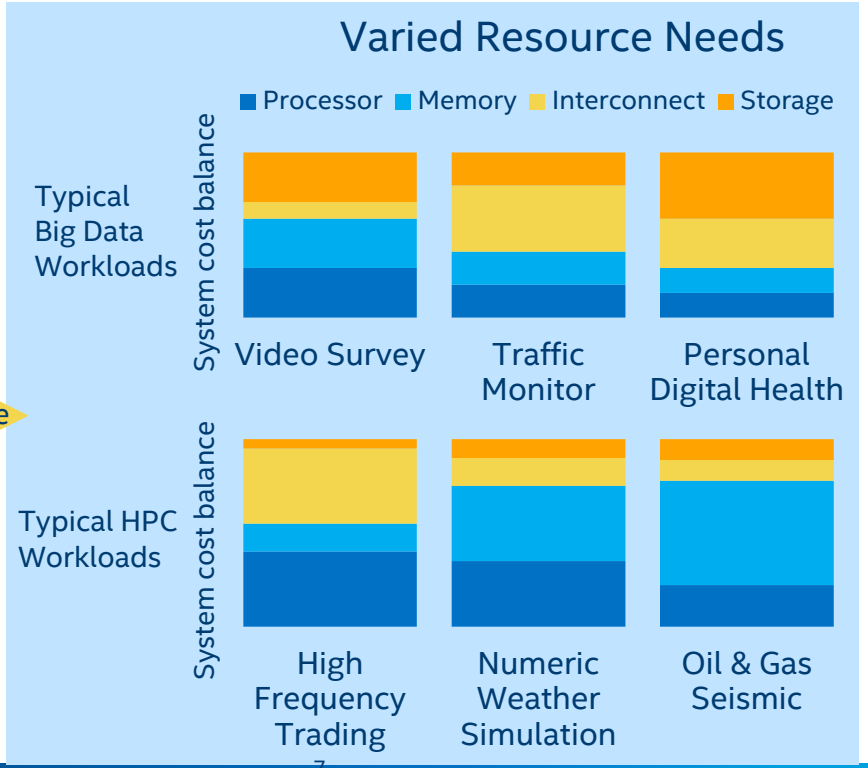
Trends in HPC + Big Data Analytics



Big Data & HPC

Ambientes de Produção





https://www.intelnervana.com/framework-optimizations/

Programação Paralela em IA/ML Vs Programação Paralela “tradicional” (HPC)

Programação Paralela em ML Frameworks

Exemplo de código

INTEL® AI TOOLS

PORTFOLIO OF SOFTWARE TOOLS TO EXPEDITE AND ENRICH AI DEVELOPMENT

TOOLKITS

Application Developers

DEEP LEARNING DEPLOYMENT

OpenVINO™†

Open Visual Inference & Neural Network Optimization toolkit for inference deployment on CPU/GPU/FPGA/VPU using TensorFlow, Caffe* & MXNet**

Intel® Movidius™ SDK

Optimized inference deployment for all Intel® Movidius™ VPUs using TensorFlow & Caffe

DEEP LEARNING — COMING SOON!

Intel® Deep Learning Studio‡

Open-source tool to compress deep learning development cycle

LIBRARIES

Data Scientists

MACHINE LEARNING LIBRARIES

Python	R	Distributed
• Scikit-learn • Pandas • NumPy	• Cart • Random Forest • e1071	• MLlib (on Spark) • Mahout

DEEP LEARNING FRAMEWORKS

Now optimized for CPU

TensorFlow · MXNet · Caffe · BigDL* (Spark)

Optimizations in progress — COMING SOON!

Caffe2 · PyTorch · CNTK · PaddlePaddle

FOUNDATION

Library Developers

ANALYTICS, MACHINE & DEEP LEARNING PRIMITIVES

Python*

Intel distribution optimized for machine learning

DAAL

Intel® Data Analytics Acceleration Library (incl machine learning)

MKL-DNN **clDNN**

Open-source deep neural network functions for CPU / integrated graphics

DEEP LEARNING GRAPH COMPILER

Intel® nGraph™ Compiler (Alpha)

Open-sourced compiler for deep learning model computations optimized for multiple devices from multiple frameworks

† Formerly the Intel® Computer Vision SDK
*Other names and brands may be claimed as the property of others.
Developer personas show above represent the primary user base for each row, but are not mutually-exclusive
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT

WRITE ONCE, DEPLOY EVERYWHERE

software.intel.com/openvino-toolkit

DEEP LEARNING

Caffe | TensorFlow | ONNX | mxnet | KALDI

- Model Optimizer
- Inference Engine
- Supports >100 Public Models, incl. 30+ Pretrained Models
- CV Algorithms

COMPUTER VISION

OpenCV* | OpenCL™ | OpenVX

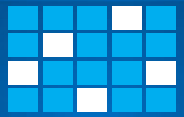
- CV Library (Kernel & Graphic APIs)
- Optimized media encode/decode functions

AGNOSTIC, COMPLEMENTARY TO MAJOR FRAMEWORKS

CROSS-PLATFORM FLEXIBILITY

HIGH PERFORMANCE, HIGH EFFICIENCY

12,000+ Developers



Over 20 Customer Products Launched based on Intel® Distribution of OpenVINO™ toolkit

Breadth of vision product portfolio

STRONG ADOPTION + RAPIDLY EXPANDING CAPABILITY

An open source version is available at 01.org/openvinotoolkit

What's Inside Intel® Distribution of OpenVINO™ toolkit

Intel® Deep Learning Deployment Toolkit

Model Optimizer
Convert & Optimize

IR

Inference Engine
Optimized Inference

30+ Pre-trained Models

Computer Vision Algorithms

Samples

IR = Intermediate Representation file

Traditional Computer Vision

Optimized Libraries & Code Samples

OpenCV*

OpenVX*

Samples

For Intel® CPU & GPU/Intel® Processor Graphics

Tools & Libraries

Increase Media/Video/Graphics Performance

Intel® Media SDK
Open Source version

OpenCL™ Drivers & Runtimes

For GPU/Intel® Processor Graphics

Optimize Intel® FPGA (Linux* only)

FPGA RunTime Environment
(from Intel® FPGA SDK for OpenCL™)

Bitstreams

OS Support: CentOS* 7.4 (64 bit), Ubuntu* 16.04.3 LTS (64 bit), Microsoft Windows* 10 (64 bit), Yocto Project* version Poky Jethro v2.0.3 (64 bit)

Intel® Architecture-Based Platforms Support

CELERON, ATOM, CORE, XEON, ARRIA 10, MOVIDIUS, IRIS Pro GRAPHICS

Intel® Vision Accelerator Design Products & AI in Production/ Developer Kits

An open source version is available at 01.org/openvinotoolkit (some deep learning functions support Intel CPU/GPU only).

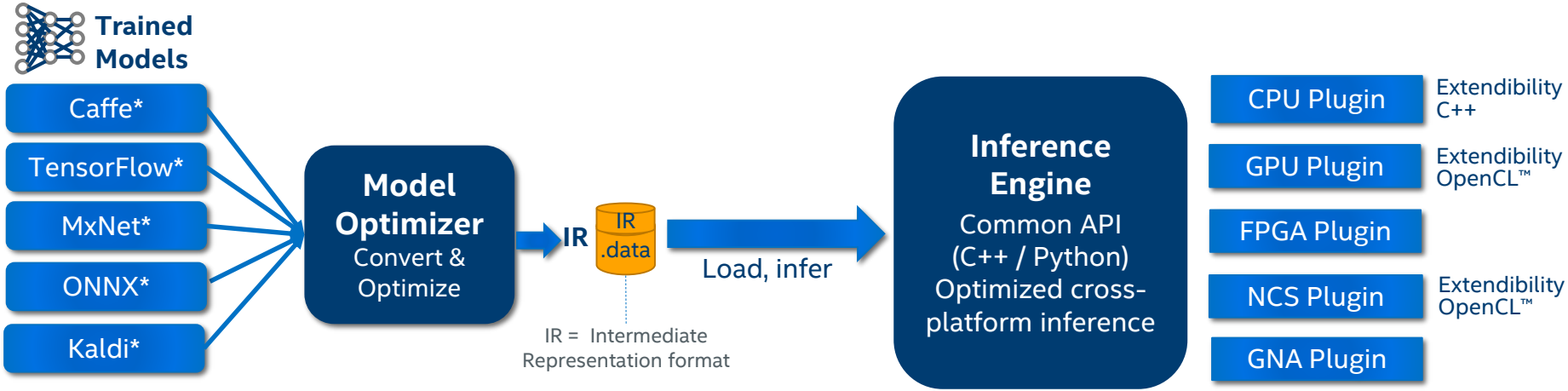
INTEL® DEEP LEARNING DEPLOYMENT TOOLKIT
FOR DEEP LEARNING INFERENCE

Model Optimizer

- **What it is**: A python based tool to import trained models and convert them to Intermediate representation.
- **Why important**: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.

Inference Engine

- **What it is**: High-level inference API
- **Why important**: Interface is implemented as dynamically loaded plugins for each hardware type. Delivers best performance for each type without requiring users to implement and maintain multiple code pathways.



GPU = Intel CPU with integrated graphics processing unit/Intel® Processor Graphics

INTEL® VISION PRODUCTS

WRITE ONCE - DEPLOY ACROSS INTEL ARCHITECTURE - LEVERAGE COMMON ALGORITHMS

OpenVINO™

Intel® CPUs
(Atom®, Core™, Xeon®)

Intel® CPUs
w/ Integrated Graphics

Intel® Movidius™ VPUs
& Intel® FPGAs

Future Accelerators
(Keem Bay, etc.)

1. Intel® Distribution of OpenVINO™ toolkit: Computer vision & deep learning inference tool with common API
2. Portfolio of hardware for computer vision & deep learning inference, device to cloud
3. Ecosystem to cover the breadth of IoT vision systems

INTEL® VISION ACCELERATOR DESIGN PRODUCTS

Add to existing Intel® architectures for accelerated DL inference capabilities

AI ON APACHE Spark™

BigDL

HIGH-PERFORMANCE
DEEP LEARNING FRAMEWORK
FOR APACHE SPARK

software.intel.com/bigdl

ANALYTICS ZOO

UNIFIED ANALYTICS + AI PLATFORM
DISTRIBUTED TENSORFLOW, KERAS AND BIGDL ON APACHE SPARK

Reference Use Cases, AI Models,
High-level APIs, Feature Engineering, etc.

https://github.com/intel-analytics/analytics-zoo

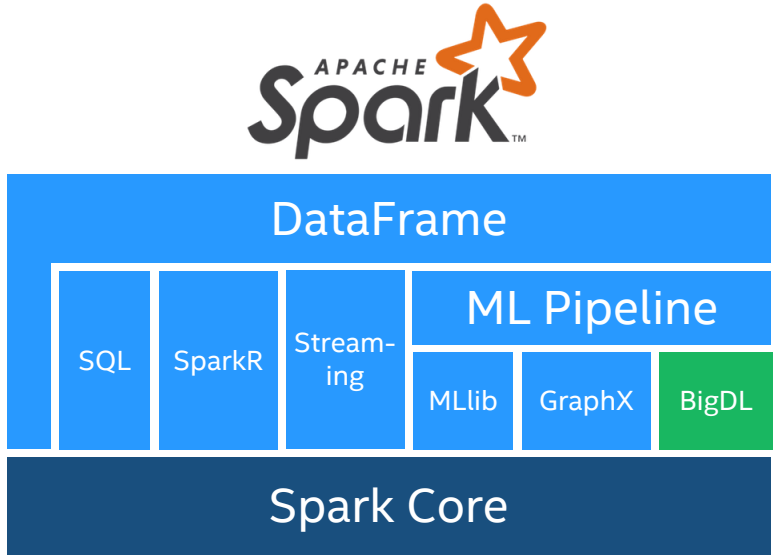
UNIFYING ANALYTICS + AI ON APACHE SPARK

BIGDL

Bringing Deep Learning to Big Data

For developers looking to run deep learning on Hadoop/Spark due to familiarity or analytics use

- **Open Sourced** Deep Learning Library for Apache Spark*
- **Make Deep learning more Accessible** to Big data users and data scientists.
- **Feature Parity** with popular DL frameworks like Caffe, Torch, Tensorflow etc.
- **Easy Customer and Developer Experience**
  - Run Deep learning Applications as Standard Spark programs;
  - Run on top of existing Spark/Hadoop clusters (No Cluster change)
- **High Performance** powered by Intel MKL and Multi-threaded programming.
- **Efficient Scale out** leveraging Spark architecture.

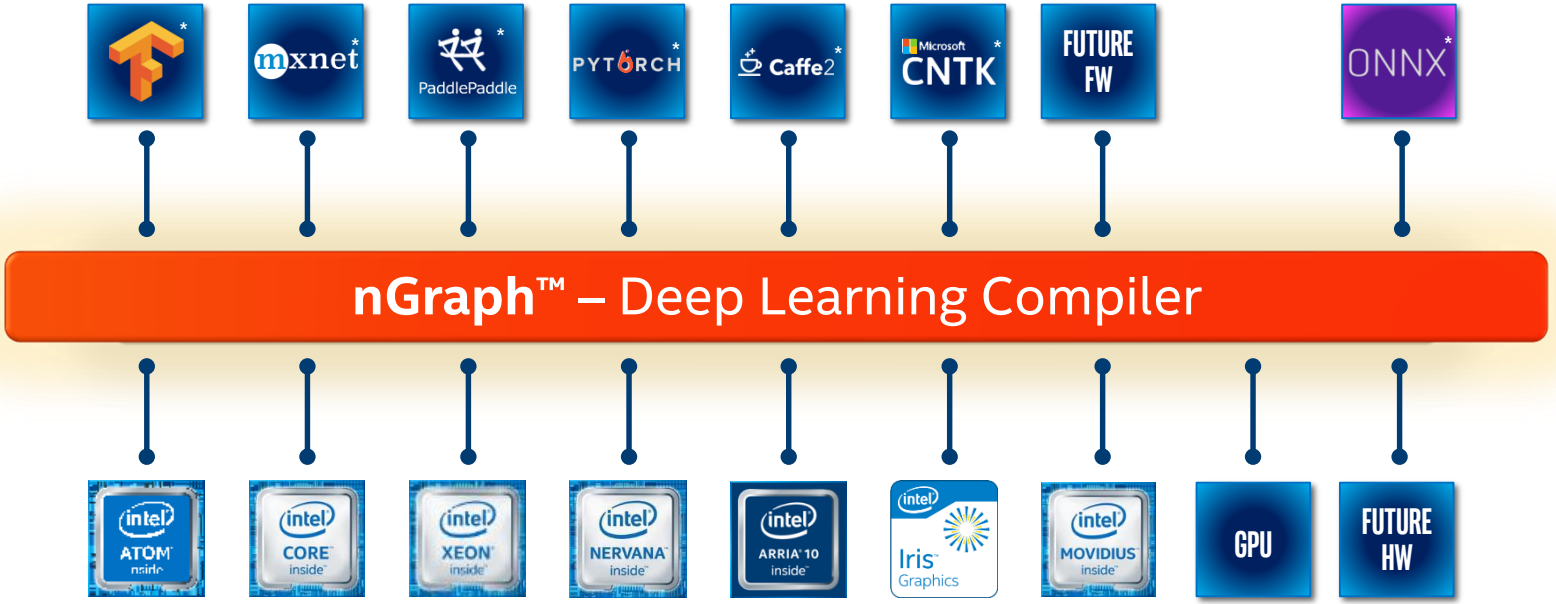


github.com/intel-analytics/BigDL

INTEL® NGRAPH™ COMPILER

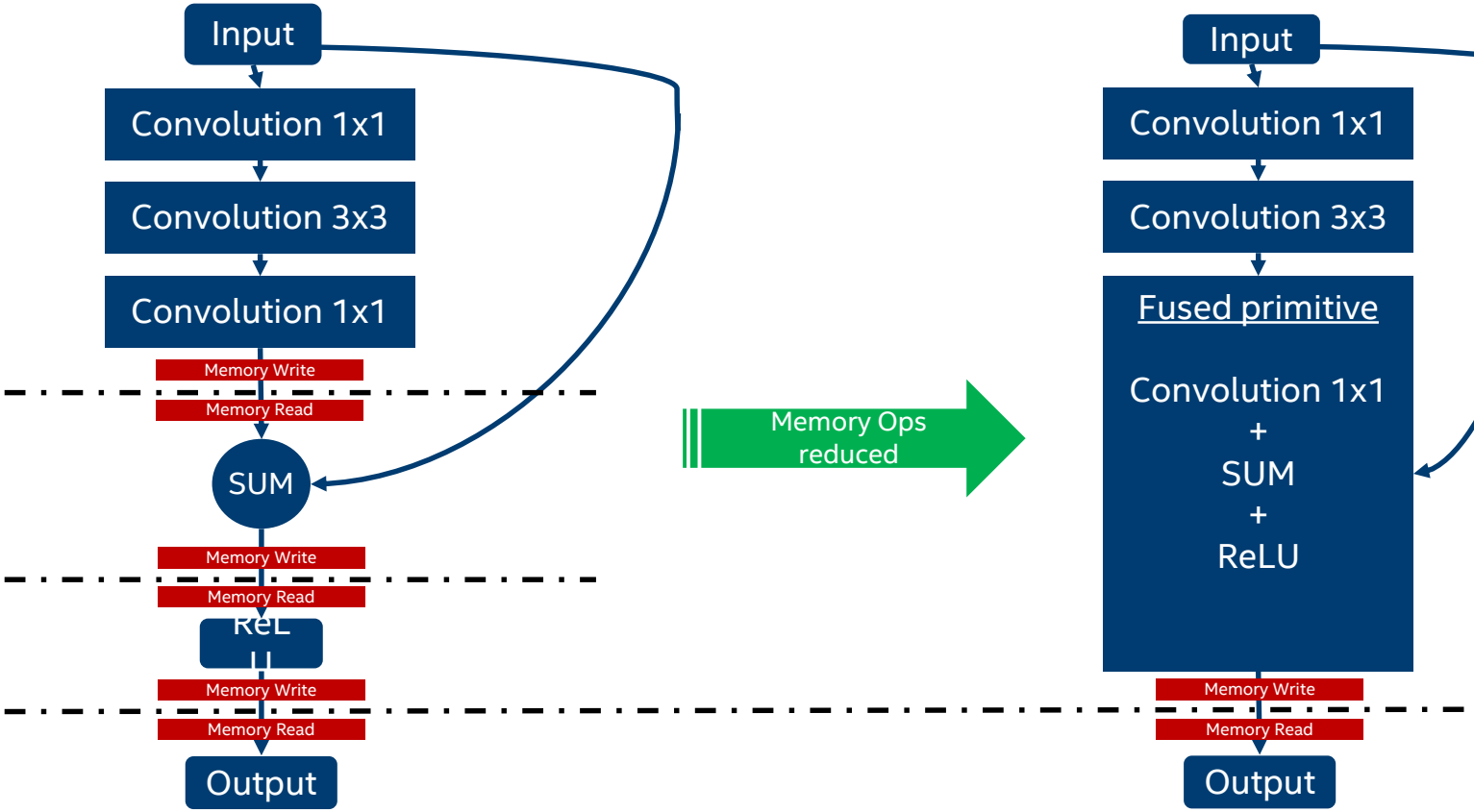
AVAILABLE IN ALPHA

Open-source compiler enabling flexibility to run models across a variety of frameworks and hardware



*Other names and brands may be claimed as the property of others.
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

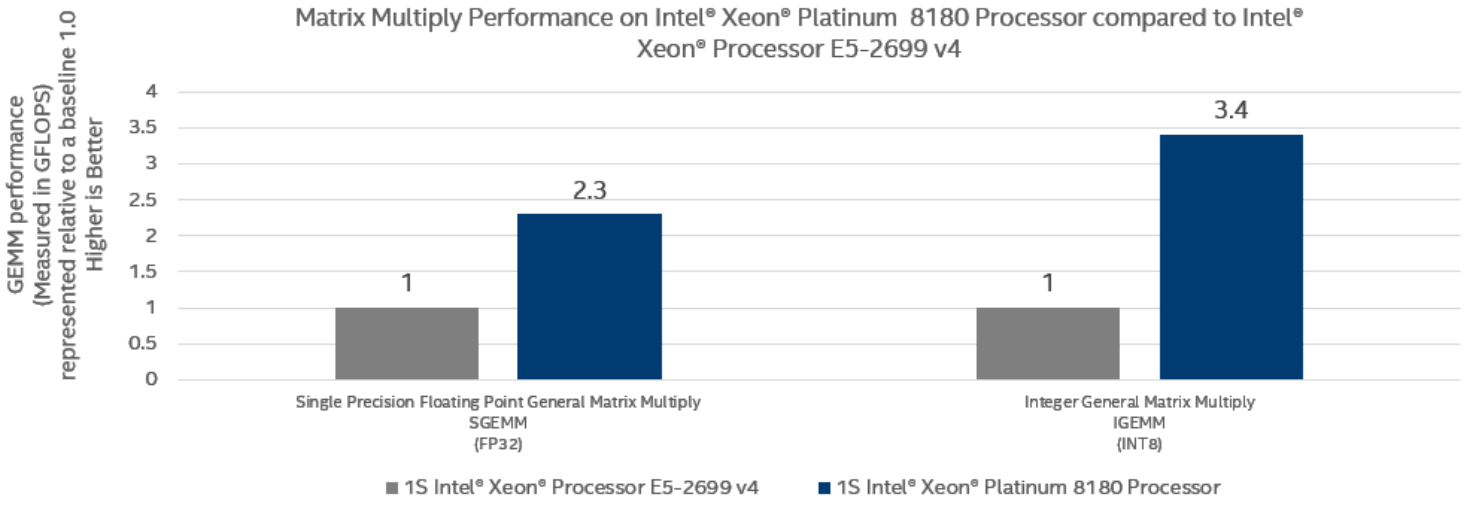
Exemplo: Layer Fusion



Integer Matrix Multiply Performance

on Intel® Xeon® Platinum 8180 Processor

Lower precision integer ops



Enhanced matrix multiply performance on Intel® Xeon® Scalable Processor

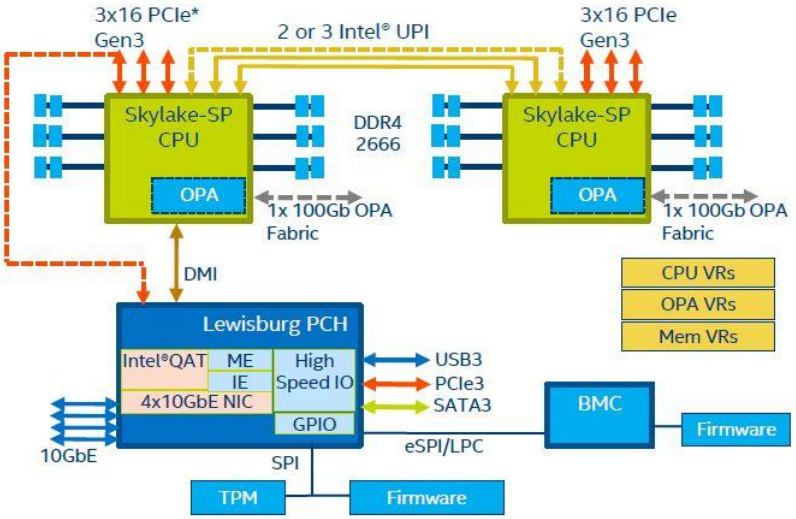
Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Configuration Details on Slide: 13

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance Source: Intel measured as of June 2017 Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Programação Paralela aplicado em IA

Intel® Xeon® Scalable Processor Feature Overview



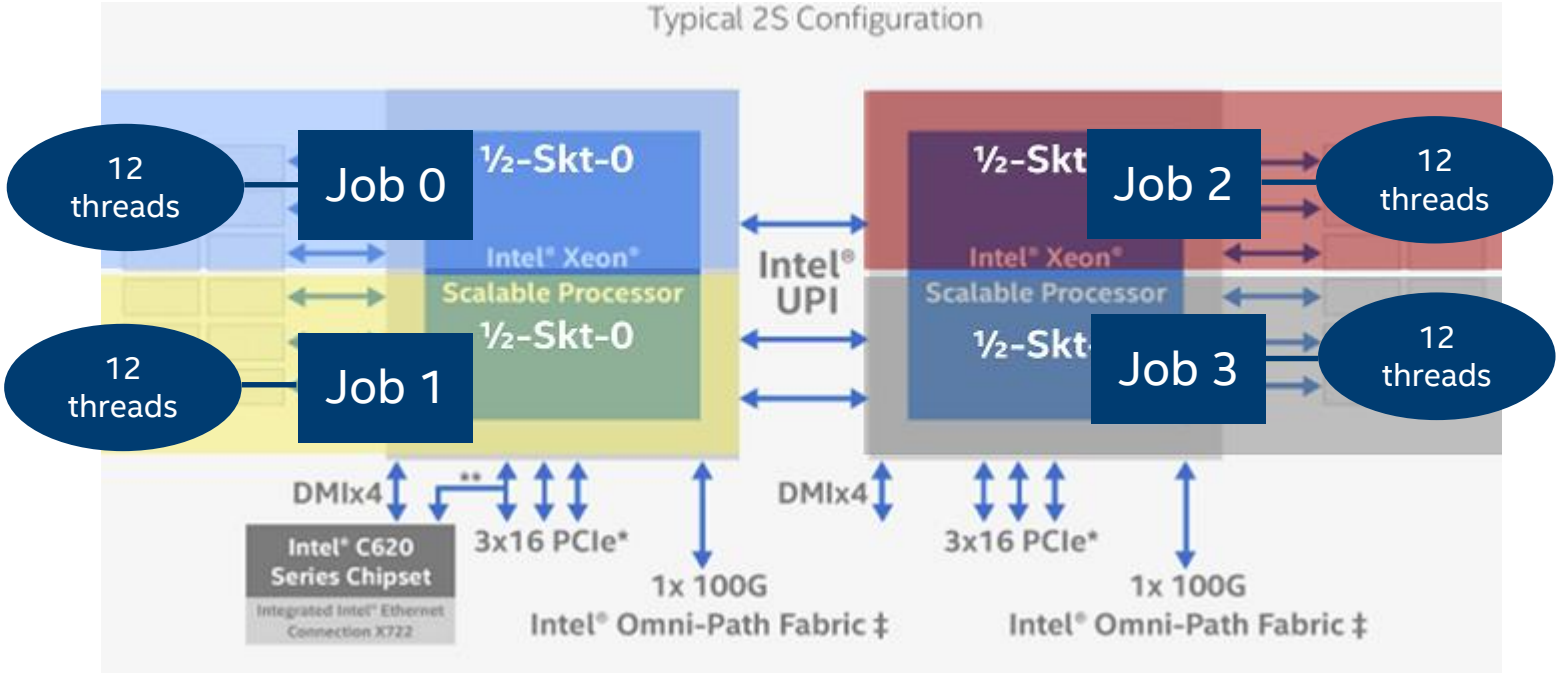
BMC: Baseboard Management Controller	PCH: Intel® Platform Controller Hub	IE: Innovation Engine
Intel® OPA: Intel® Omni-Path Architecture	Intel QAT: Intel® QuickAssist Technology	ME: Manageability Engine
NIC: Network Interface Controller	VMD: Volume Management Device	NTB: Non-Transparent Bridge

Feature	Details
Socket	Socket P
Scalability	2S, 4S, 8S, and >8S (with node controller support)
CPU TDP	70W – 205W
Chipset	Intel® C620 Series (code name Lewisburg)
Networking	Intel® Omni-Path Fabric (integrated or discrete) 4x10GbE (integrated w/ chipset) 100G/40G/25G discrete options
Compression and Crypto Acceleration	Intel® QuickAssist Technology to support 100Gb/s comp/decomp/crypto 100K RSA2K public key
Storage	Integrated QuickData Technology, VMD, and NTB Intel® Optane™ SSD, Intel® 3D-NAND NVMe & SATA SSD
Security	CPU enhancements (MBE, PPK, MPX) Manageability Engine Intel® Platform Trust Technology Intel® Key Protection Technology
Manageability	Innovation Engine (IE) Intel® Node Manager Intel® Datacenter Manager

Programação Paralela aplicado em IA

Técnicas de HPC aplicadas para IA

SERPRO
Serviço Federal de Processamento de Dados



`libnumactl` `kmp_affinity`

https://software.intel.com/en-us/articles/boosting-deep-learning-training-inference-performance-on-xeon-and-xeon-phi

Centros de Excelência em Inteligência Artificial - Intel

Casos de sucesso

SERPRO
Serviço Federal de Processamento de Dados

Thiago Oliveira, superintendente de Engenharia de Infraestrutura do SERPRO

"Validador Cognitivo de Infrações de Trânsito"

✓ Performance 22.5x mais rápida em "Xeon Scalable Processors"

"...um processamento de multas que antes levava 45 horas agora poderá ser realizado em menos de 2 horas."

✓ Desenvolvimento do modelo matemático

"Com isso, tivemos uma acurácia de 90% no sistema, além da automação de todo o projeto",
disse Gustavo Rocha, chefe de divisão do SERPRO,"

TensorFlow for CPU

Aplicando técnica “Afinidade de Processos” (NUMA aware) no TensorFlow

intra_op_parallelism_threads: Nodes that can use multiple threads to parallelize their execution will schedule the individual pieces into this pool.

inter_op_parallelism_threads: All ready nodes are scheduled in this pool.

```
config = tf.ConfigProto()
config.intra_op_parallelism_threads = 44
config.inter_op_parallelism_threads = 44
tf.Session(config=config)
```

Source: https://www.tensorflow.org/guide/performance/overview#optimizing_for_cpu

Programação Paralela em ML Vs Programação Paralela (HPC)

Programação Paralela em ML Frameworks

Exemplo de código

Programação Paralela aplicado em IA

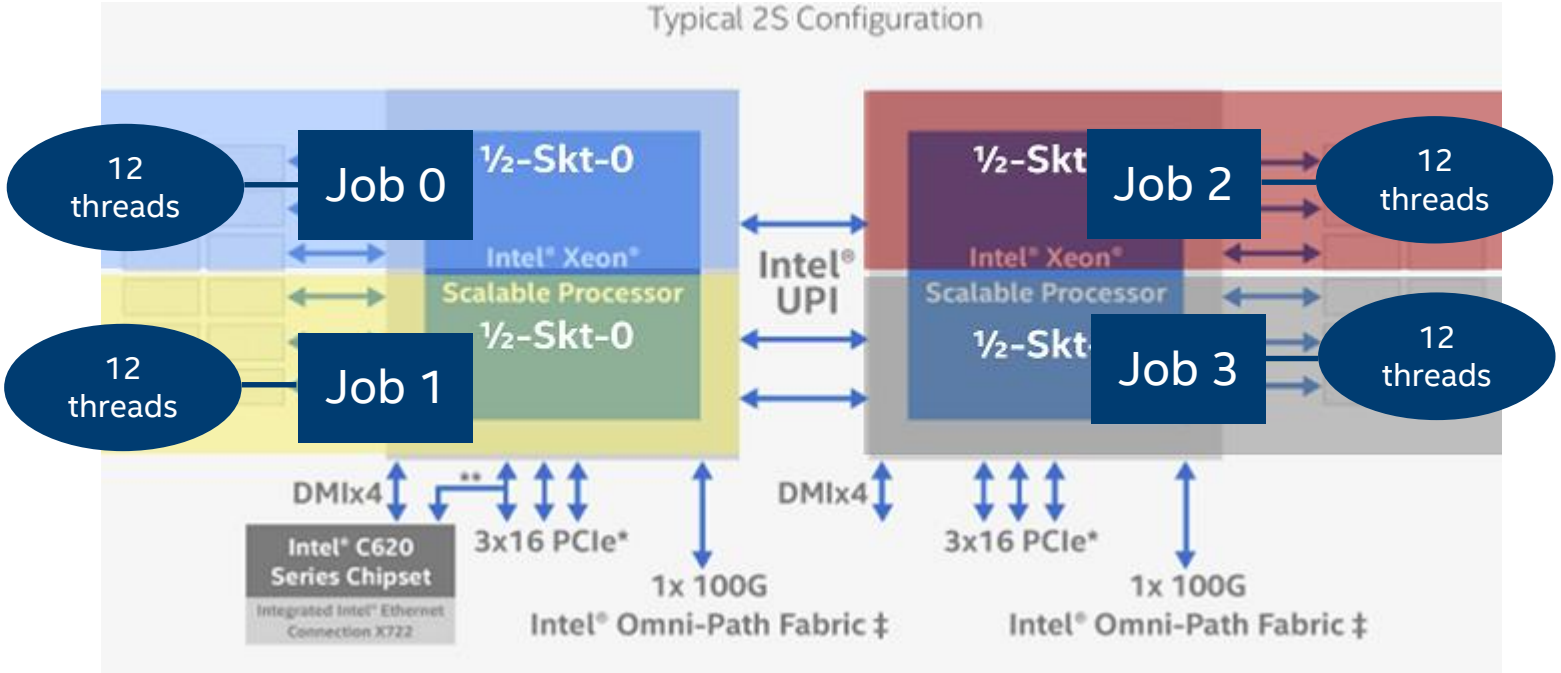
```
CPU op-mode(s):          32-bit, 64-bit
Byte Order:              Little Endian
CPU(s):                  64
On-line CPU(s) list:     0-63
Thread(s) per core:      2
Core(s) per socket:      16
Socket(s):               2
NUMA node(s):            2
Vendor ID:               GenuineIntel
CPU family:              6
Model:                   85
Model name:              Intel(R) Xeon(R) Platinum 8153 CPU @ 2.00GHz
Stepping:                4
CPU MHz:                 1000.156
CPU max MHz:             2800.0000
CPU min MHz:             1000.0000
BogoMIPS:                4000.00
Virtualization:          VT-x
L1d cache:               32K
L1i cache:               32K
L2 cache:                1024K
L3 cache:                22528K
NUMA node0 CPU(s):       0-15,32-47
NUMA node1 CPU(s):       16-31,48-63
Flags:                   fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
 arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq d
tes64 monitor ds_cpl vmx smx est tm2 ssse3 fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic mo
vbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch epb cat_l3 c
dp_l3 invpcid_single intel_pt spec_ctrl ibpb_support tpr_shadow vnmi flexpriority ept vpid fsg
sbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm mpx rdt_a avx512f avx512dq rdsee
d adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 cqm_llc cqm_occu
p_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts hwp hwp_act_window hwp_epp hwp_pkg_r
eq
```

Entendendo o ambiente:

- *Dual socket*
- *AVX-512*
- *16 cores / socket*
- *32 threads / socket*
- *Total: 64 threads*

Programação Paralela aplicada em IA

Técnicas de HPC aplicadas para IA



`libnumactl` `kmp_affinity`

https://software.intel.com/en-us/articles/boosting-deep-learning-training-inference-performance-on-xeon-and-xeon-phi

Programação Paralela aplicado em IA

```
def model(data, train=False):
    """The Model definition."""
    # 2D convolution, with 'SAME' padding (i.e. the output feature map has
    # the same size as the input). Note that {strides} is a 4D array whose
    # shape matches the data layout: [image index, y, x, depth].
    conv = tf.nn.conv2d(data,
                        conv1_weights,
                        strides=[1, 1, 1, 1],
                        padding='SAME')

    # Bias and rectified linear non-linearity.
    relu = tf.nn.relu(tf.nn.bias_add(conv, conv1_biases))

    # Max pooling. The kernel size spec ksize also follows the layout of
    # the data. Here we have a pooling window of 2, and a stride of 2.
    pool = tf.nn.max_pool(relu,
                          ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1],
                          padding='SAME')
    conv = tf.nn.conv2d(pool,
                        conv2_weights,
                        strides=[1, 1, 1, 1],
                        padding='SAME')
    relu = tf.nn.relu(tf.nn.bias_add(conv, conv2_biases))
    pool = tf.nn.max_pool(relu,
                          ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1],
                          padding='SAME')

    # Reshape the feature map cuboid into a 2D matrix to feed it to the
    # fully connected layers.
    pool_shape = pool.get_shape().as_list()
    reshape = tf.reshape(
        pool,
        [pool_shape[0], pool_shape[1] * pool_shape[2] * pool_shape[3]])

    # Fully connected layer. Note that the '+' operation automatically
    # broadcasts the biases.
    hidden = tf.nn.relu(tf.matmul(reshape, fc1_weights) + fc1_biases)

    # Add a 50% dropout during training only. Dropout also scales
    # activations such that no rescaling is needed at evaluation time.
    if train:
        hidden = tf.nn.dropout(hidden, 0.5, seed=SEED)
    return tf.matmul(hidden, fc2_weights) + fc2_biases

print('Done')
```

Codigo de demonstração:
MNIST

Topologia:
Convolution + reLu + maxPool +
Convolution + reLu + maxPool

Programação Paralela aplicado em IA

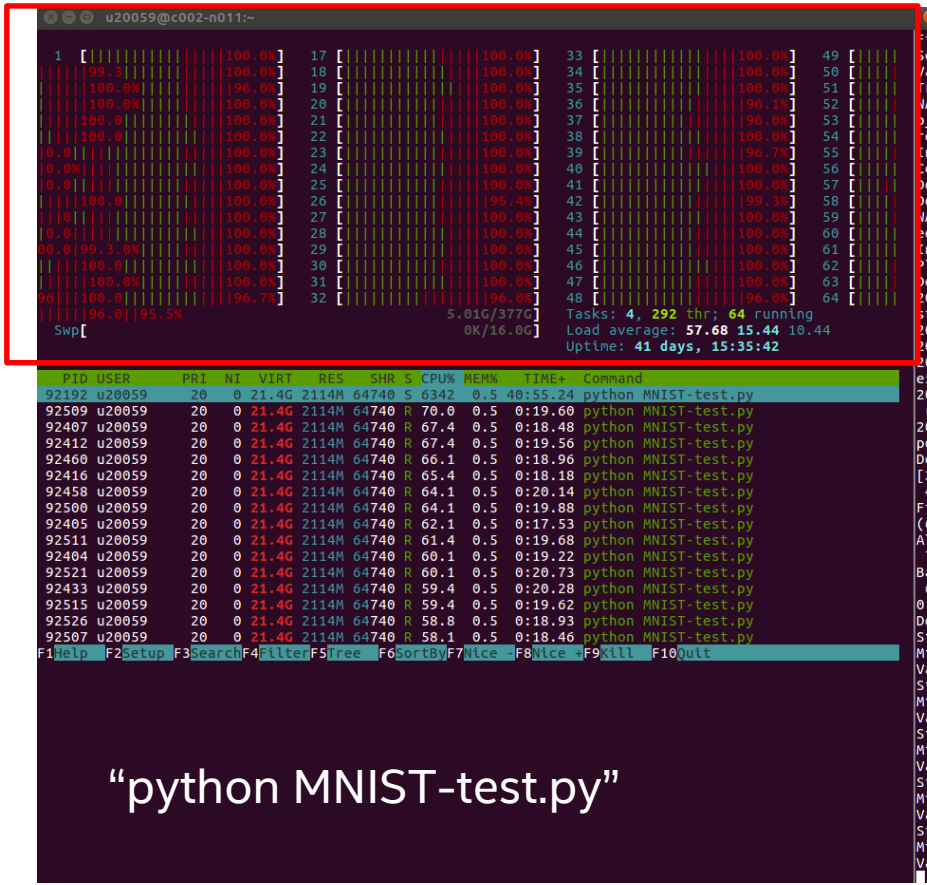
```
[u20059@c002-n011 demo-tdc]$ conda env list
# conda environments:
#
base                  *  /glob/intel-python/versions/2018u2/intelpython3
ng                       /home/u20059/.conda/envs/ng
ng-compiled              /home/u20059/.conda/envs/ng-compiled
ngraph-compiled          /home/u20059/.conda/envs/ngraph-compiled
tf-conda                 /home/u20059/.conda/envs/tf-conda
tf-pip                   /home/u20059/.conda/envs/tf-pip

[u20059@c002-n011 demo-tdc]$ conda create -n tf-pip-2
Solving environment: done

## Package Plan ##

  environment location: /home/u20059/.conda/envs/tf-pip-2


Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate tf-pip-2
#
# To deactivate an active environment, use
#
#     $ conda deactivate

[u20059@c002-n011 demo-tdc]$
```

- Preparando o ambiente via Anaconda

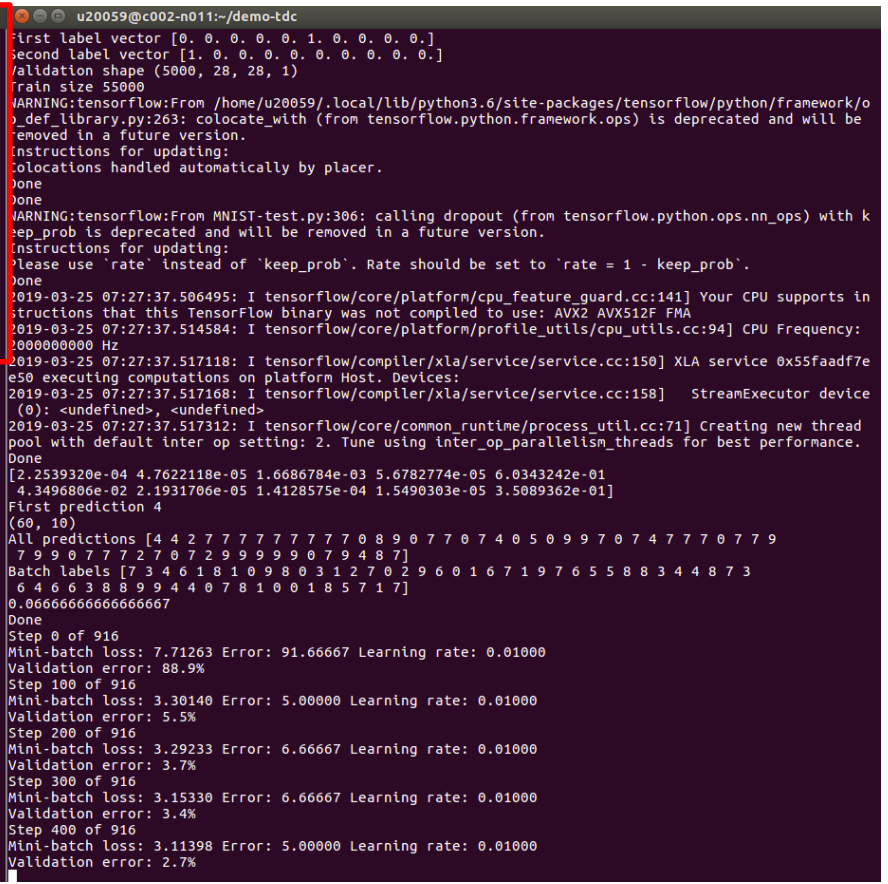
"conda create –n tf-pip-2"

"pip install intel-tensorflow"

Programação Paralela aplicado em IA



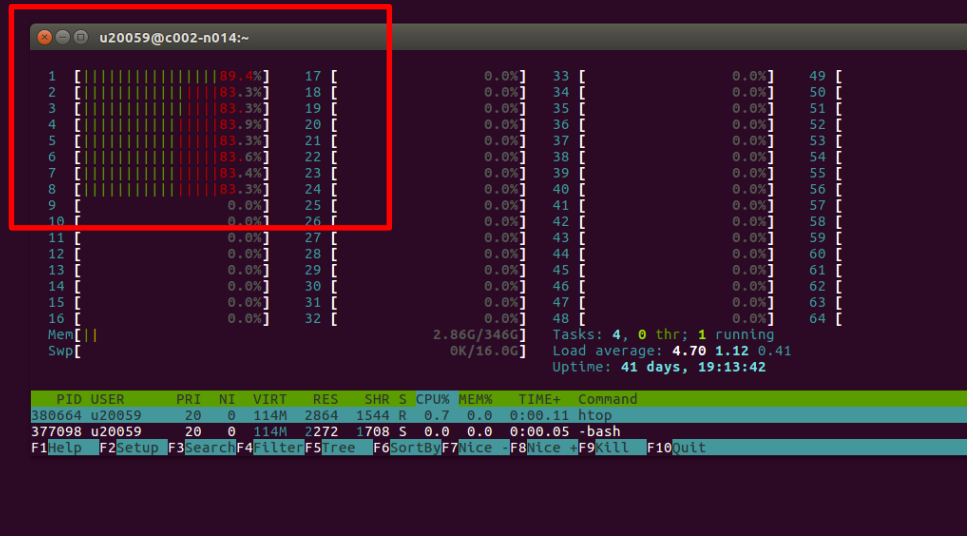
“python MNIST-test.py”



Programação Paralela aplicado em IA

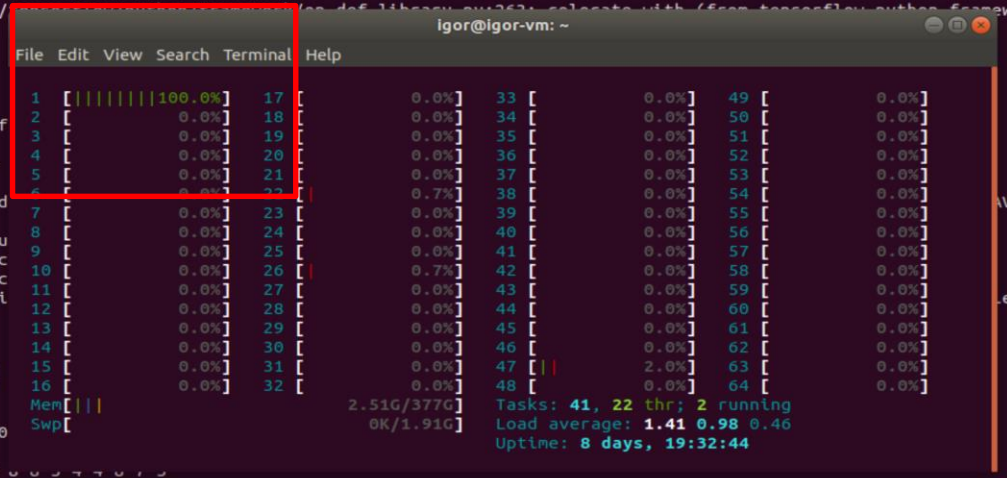
```
2019-03-25 11:06:39.927199: I tensorflow/core/common_runtime/process_util.cc:71] Creating new thread pool with default inter op setting: 2. Tune using inter_op_parallelism_threads for best performance.
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380696 thread 8 bound to OS proc set 0
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380695 thread 9 bound to OS proc set 1
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380710 thread 10 bound to OS proc set 2
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380711 thread 11 bound to OS proc set 3
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380712 thread 12 bound to OS proc set 4
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380714 thread 14 bound to OS proc set 6
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380713 thread 13 bound to OS proc set 5
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380715 thread 15 bound to OS proc set 7
OMP: Info #250: KMP_AFFINITY: pid 380671 tid 380716 thread 16 bound to OS proc set 0
Done
[2.2539320e-04 4.7622118e-05 1.6686784e-03 5.6782774e-05 6.0343242e-01
 4.3496806e-02 2.1931706e-05 1.4128575e-04 1.5490303e-05 3.5089362e-01]
First prediction 4
(60, 10)
All predictions [4 4 2 7 7 7 7 7 7 7 7 7 0 8 9 0 7 7 0 7 4 0 5 0 9 9 7 0 7 4 7 7 7 0 7 7 9
 7 9 9 0 7 7 7 2 7 0 7 2 9 9 9 9 9 0 7 9 4 8 7]
Batch labels [7 3 4 6 1 8 1 0 9 8 0 3 1 2 7 0 2 9 6 0 1 6 7 1 9 7 6 5 5 8 8 3 4 4 8 7 3
 6 4 6 6 3 8 8 9 9 4 4 0 7 8 1 0 0 1 8 5 7 1 7]
0.06666666666666667
Done
Step 0 of 916
Mini-batch loss: 7.71263 Error: 91.66667 Learning rate: 0.01000
Validation error: 88.9%
Step 100 of 916
Mini-batch loss: 3.31044 Error: 8.33333 Learning rate: 0.01000
Validation error: 5.8%
Step 200 of 916
Mini-batch loss: 3.30167 Error: 8.33333 Learning rate: 0.01000
Validation error: 3.8%
Step 300 of 916
Mini-batch loss: 3.12416 Error: 3.33333 Learning rate: 0.01000
Validation error: 3.3%
Step 400 of 916
Mini-batch loss: 3.09580 Error: 5.00000 Learning rate: 0.01000
Validation error: 2.7%
Step 500 of 916
Mini-batch loss: 3.02841 Error: 1.66667 Learning rate: 0.01000
Validation error: 2.4%
Step 600 of 916
Mini-batch loss: 3.05534 Error: 5.00000 Learning rate: 0.01000
Validation error: 2.0%
Step 700 of 916
Mini-batch loss: 3.15112 Error: 8.33333 Learning rate: 0.01000
Validation error: 2.2%
Step 800 of 916
Mini-batch loss: 3.05624 Error: 5.00000 Learning rate: 0.01000
Validation error: 1.9%
Step 900 of 916
Mini-batch loss: 2.85126 Error: 0.00000 Learning rate: 0.01000
Validation error: 2.0%
Walltime: 28.40896463394165
Test error: 2.1%
(tf-pip-2) [u20059@c002-n014 demo-tdc]$
```

"numactl -C 0-7 python MNIST-test.py"



Programação Paralela aplicado em IA

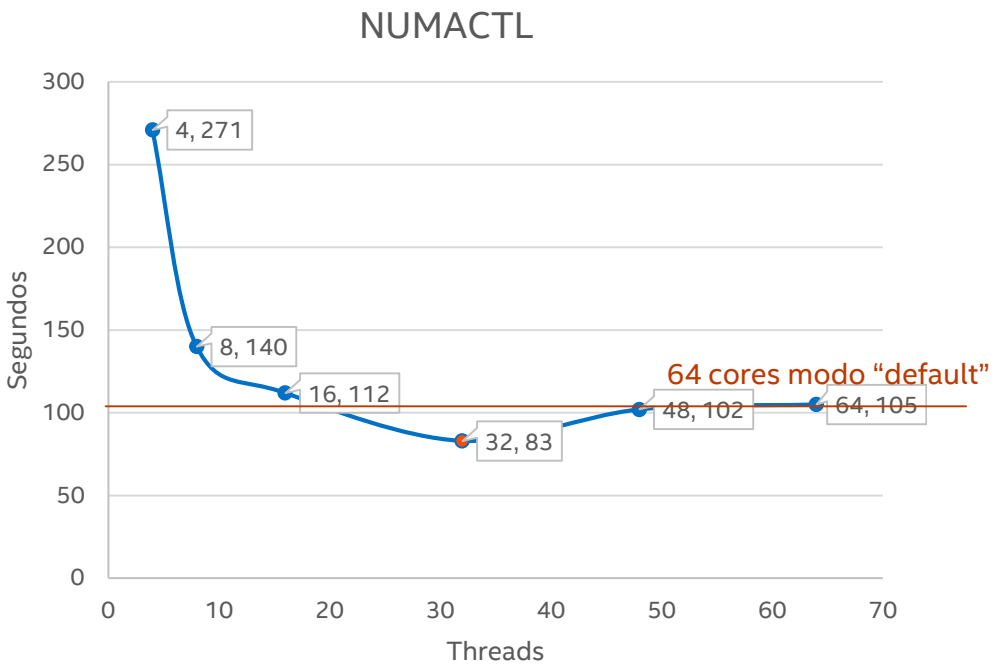
```
Extracting /tmp/mnist-data/train-labels-idx1-ubyte.gz
Extracting /tmp/mnist-data/t10k-labels-idx1-ubyte.gz
Training labels shape (60000, 10)
First label vector [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
Second label vector [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Validation shape (5000, 28, 28, 1)
Train size 55000
WARNING:tensorflow:From /home/u20059/.local/lib/python3.6/site-packages/
s deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
Done
Done
WARNING:tensorflow:From MNIST-test.py:306: calling dropout (from tensorf
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
Done
2019-04-08 12:37:20.546485: I tensorflow/core/platform/cpu_feature_guard
 FMA
2019-04-08 12:37:20.571720: I tensorflow/core/platform/profile_utils/cpu
2019-04-08 12:37:20.571967: I tensorflow/compiler/xla/service/service.cc
2019-04-08 12:37:20.571996: I tensorflow/compiler/xla/service/service.cc
2019-04-08 12:37:20.572149: I tensorflow/core/common_runtime/process_uti
ds for best performance.
Done
[2.2539249e-04 4.7621914e-05 1.6686745e-03 5.6782646e-05 6.0343218e-01
 4.3496739e-02 2.1931655e-05 1.4128556e-04 1.5490252e-05 3.5089380e-01]
First prediction 4
(60, 10)
All predictions [4 4 2 7 7 7 7 7 7 7 7 7 0 8 9 0 7 7 0 7 4 0 5 0 9 9 7 0
 7 9 9 0 7 7 7 2 7 0 7 2 9 9 9 9 9 0 7 9 4 8 7]
Batch labels [7 3 4 6 1 8 1 0 9 8 0 3 1 2 7 0 2 9 6 0 1 6 7 1 9 7 6 5 5
 6 4 6 6 3 8 8 9 9 4 4 0 7 8 1 0 0 1 8 5 7 1 7]
0.06666666666666667
Done
Step 0 of 916
Mini-batch loss: 7.71263 Error: 91.66667 Learning rate: 0.01000
Validation error: 88.9%
Step 100 of 916
Mini-batch loss: 3.30302 Error: 8.33333 Learning rate: 0.01000
Validation error: 5.7%
Step 200 of 916
Mini-batch loss: 3.33852 Error: 8.33333 Learning rate: 0.01000
Validation error: 3.7%
Step 300 of 916
Mini-batch loss: 3.12427 Error: 1.66667 Learning rate: 0.01000
Validation error: 3.3%
```



“numactl -C 0 python MNIST-test.py”

Programação Paralela aplicado em IA

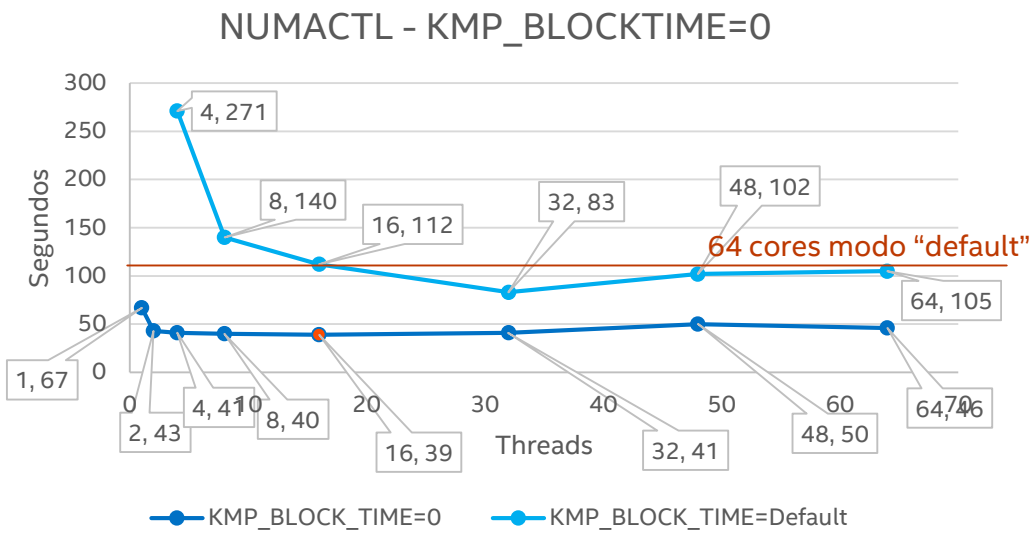
Tempo para 64 Threads “default”: 102 segundos



numactl –C 0-15,16-31 python MNIST.py

- Mais cores não significa maior performance
- 48 threads teve mesma performance que 64 threads (102s)
- Melhor tempo com 32 threads (83s) – 1.22x speedup

Programação Paralela aplicado em IA

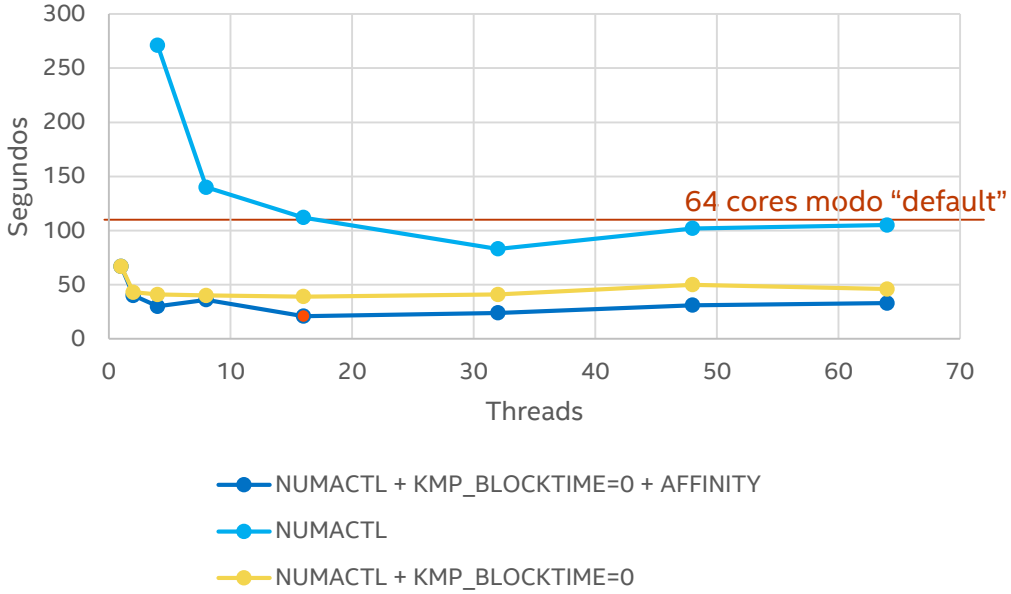



```
export KMP_BLOCKTIME=0
numactl –C 0-15,16-31 python MNIST.py
```

- KMP_BLOCKTIME: *tempo em milisegundos de espera da thread, após executar sua tarefa, antes de dormir*
- 2.68x speedup
- Melhor tempo com 16 threads
- Melhor Performance x benefício com 2 Threads

Programação Paralela aplicado em IA

```
export KMP_BLOCKTIME=0
export KMP_AFFINITY=granularity=fine,verbose,compact,1,0
numactl –C 0-15 python MNIST.py
```



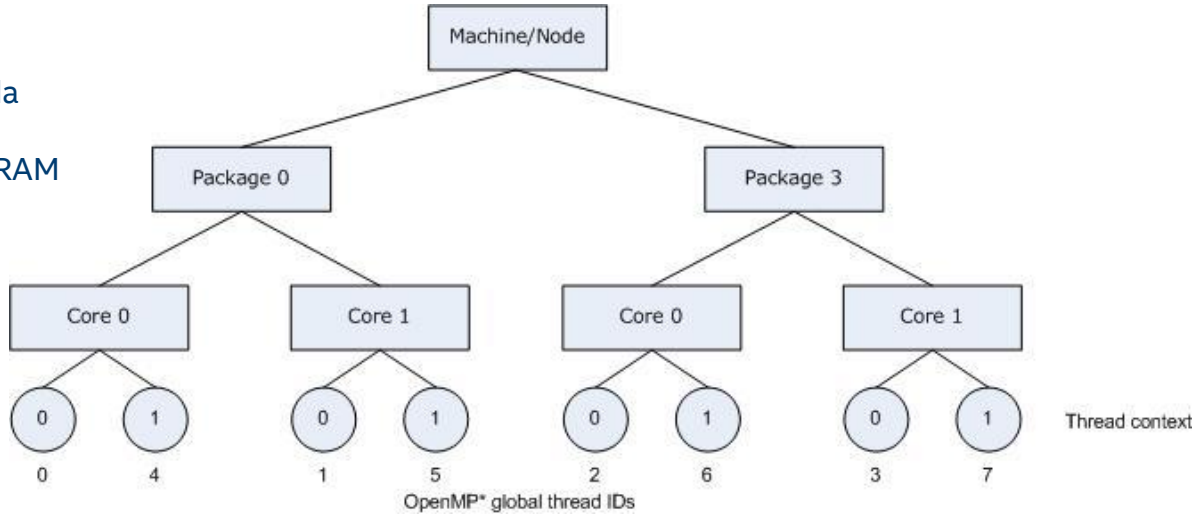
- **16 threads : 4.86x speedup !**
  - Menor custo de infra-estrutura
  - Mais jobs de treinamento ao mesmo tempo
  - Modelos maiores
- Sem alteração de código

Programação Paralela aplicado em IA

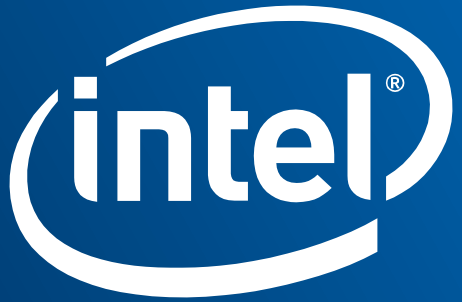
KMP_AFFINITY=granularity=fine,verbose,compact,1,0

- Como as Threads são distribuídas entre os Cores e Sockets
- Impacta bandwidth: “velocidade de memória”

- Compact:
  - Threads próximas entre si
  - Troca de dados entre elas mais rápida
  - Dados cabem na cache,
  - Pouca troca de dados entre CPU e DRAM



Obrigado !



Dúvidas

BACKUP

What's New in Intel® Distribution of OpenVINO™ toolkit
2018 R5

- **Extends neural network support** to include LSTM (long short-term memory) from ONNX*, TensorFlow*& MXNet* frameworks, & 3D convolutional-based networks in preview mode (CPU-only) for **non-vision use cases**.
- **Introduces Neural Network Builder API** (preview), providing flexibility to create a graph from simple API calls and directly deploy via the Inference Engine.
- **Improves Performance** - Delivers significant CPU performance boost on multicore systems through **new parallelization techniques via streams**. Optimizes performance on Intel® Xeon®, Core™ & Atom processors through **INT8-based primitives** for Intel® Advanced Vector Extensions (Intel® AVX-512), Intel® AVX2 & SSE4.2.
- **Supports Raspberry Pi* hardware** as a host for the Intel® Neural Compute Stick 2 (preview). Offload your deep learning workloads to this low-cost, low-power USB.
- **Adds 3 new optimized pretrained models** (for a total of 30+): Text detection of indoor/outdoor scenes, and 2 single-image super resolution networks that enhance image resolution by a factor of 3 or 4.

See product site & release notes for more details about 2018 R4.

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com]..

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

The cost reduction scenarios described are intended to enable you to get a better understanding of how the purchase of a given Intel based product, combined with a number of situation-specific variables, might affect future costs and savings. Circumstances will vary and there may be unaccounted-for costs related to the use and deployment of a given product. Nothing in this document should be interpreted as either a promise of or contract for a given level of costs or cost reduction.

Software and workloads used in performance tests may have been optimized for performance only on Intel® microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit intel.com/performance.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit intel.com/benchmarks.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Benchmark Segment	AI/ML/DL
Benchmark type	Training
Benchmark Metric	Training Throughput (images/sec)
Framework	BigDL master trunk with Spark 2.1.1
Topology	Inception V1, VGG, ResNet-50, ResNet-152
# of Nodes	8, 16 (multiple configurations)
Platform	Purley
Sockets	2S
Processor	Intel ® Xeon ® Scalable Platinum 8180 Processor (Skylake): 28-core @ 2.5 GHz (base), 3.8 GHz (max turbo), 205W Intel ® Xeon ® Processor E5-2699v4 (Broadwell): 22-core @ 2.2 GHz (base), 3.6 GHz (max turbo), 145W
Enabled Cores	Skylake: 56 per node, Broadwell: 44 per node
Total Memory	Skylake: 384 GB, Broadwell: 256 GB
Memory Configuration	Skylake: 12 slots * 32 GB @ 2666 MHz Micron DDR4 RDIMMs Broadwell: 8 slots * 32 GB @ 2400 MHz Kingston DDR4 RDIMMs
Storage	Skylake: Intel® SSD DC P3520 Series (2TB, 2.5in PCIe 3.0 x4, 3D1, MLC) Broadwell: 8 * 3 TB Seagate HDDs
Network	1 * 10 GbE network per node

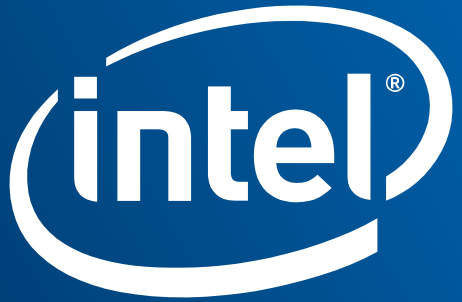
OS	CentOS Linux reléase 7.3.1611 (Core), Linux kernel 4.7.2.el7.x86_64
HT	On
Turbo	On
Computer Type	Dual-socket server
Framework Version	https://github.com/intel-analytics/BigDL
Topology Version	https://github.com/google/inception
Dataset, version	ImageNet, 2012; Cifar-10
Performance command (Inception v1)	spark-submit --class com.intel.analytics.bigdl.models.inception.TrainInceptionV1 --master spark://$master_hostname:7077 --executor-cores=36 --num-executors=16 --total-executor-cores=576 --driver-memory=60g --executor-memory=300g $BIGDL_HOME/dist/lib/bigdl-*-SNAPSHOT-jar-with-dependencies.jar --batchSize 2304 --learningRate 0.0896 -f hdfs:///user/root/sequence/ --checkpoint $check_point_folder
Data setup	Data was stored on HDFS and cached in memory before training
Java	JDK 1.8.0 update 144
MKL Library version	Intel MKL 2017

SPARK SQL CONFIGURATIONS

		AEP	DRAM
Hardware	DRAM	192GB (12x 16GB DDR4)	768GB (24x 32GB DDR4)
	Apache Pass	1TB (ES2: 8 x 128GB)	N/A
	AEP Mode	App Direct (Memkind)	N/A
	SSD	N/A	N/A
	CPU	Worker: Intel® Xeon® Platinum 8170 @ 2.10GHz (Thread(s) per core: 2, Core(s) per socket: 26, Socket(s): 2 CPU max MHz: 3700.0000 CPU min MHz: 1000.0000 L1d cache: 32K, L1i cache: 32K, L2 cache: 1024K, L3 cache: 36608K)	
	OS	4.16.6-202.fc27.x86_64 (BKC: WW26, BIOS: SE5C620.86B.01.00.0918.062020181644)	
Software	OAP	1TB AEP based OAP cache	620GB DRAM based OAP cache
	Hadoop	8 * HDD disk (ST1000NX0313, 1-replica uncompressed & plain encoded data on Hadoop)	
	Spark	1 * Driver (5GB) + 2 * Executor (62 cores, 74GB), spark.sql.oap.rowgroup.size=1MB	
	JDK	Oracle JDK 1.8.0_161	
Workload	Data Scale	2.6TB (9 queries related data is of 729.4GB in capacity)	
	TPC-DS Queries	9 I/O intensive queries (Q19,Q42,Q43,Q52,Q55, Q63,Q68,Q73,Q98)	
	Multi-Tenants	9 threads (Fair scheduled)	

APACHE CASSANDRA CONFIGURATIONS

			NVMe	Apache Pass
Server	Hardware	System Details	Intel® Server Board Purely Platform (2 socket)	
		CPU	Dual Intel® Xeon® Platinum 8180 Processors, 28 core/socket, 2 sockets, 2 threads per core	
		Hyper-Threading	Enabled	
		DRAM	DDR4 dual rank 192GB total = 12 DIMMs 16GB@2667Mhz	DDR4 dual rank 384GB total = 12 DIMMs 32GB@2667Mh
		Apache Pass	N/A	AEP ES.2 1.5TB total = 12 DIMMs * 128GB Capacity each: Single Rank, 128GB, 15W
		Apache Pass Mode	N/A	App-Direct
		NVMe	4 x Intel P3500 1.6TB NVMe devices	N/A
		Network	10Gbit on board Intel NIC	
	Software	OS	Fedora 27	
		Kernel	Kernel: 4.16.6-202.fc27.x86_64	
		Cassandra Version	3.11.2 release	Cassandra 4.0 trunk, with App Direct patch version 2.1, software found at https://github.com/shyla226/cassandra/tree/13981 with PCJ library: https://github.com/pmem/pcj
		JDK	Oracle Hotspot JDK (JDK1.8 u131)	
		Spectra/Meltdown Compliant	Patched for variants 1/2/3	
	Cassandra Parameters	Number of Cassandra Instances	1	14
		Cluster Nodes	One per Cluster	
		Garbage Collector	CMS	Parallel
		JVM Options (difference from default)	-Xms64G -Xmx64G	-Xms20G -Xmx20G -Xmn8G -XX:+UseAdaptiveSizePolicy -XX:ParallelGCThreads=5
		Schema	cqlstress-insanity-example.yaml	
		DataBase Size per Instance	1.25 Billion entries	100 K entries
Client(s)	Hardware	Number of Client machines	1	2
		System	Intel® Server Board model S2600WFT (2 socket)	
		CPU	Dual Intel® Xeon® Platinum 8176M CPU @ 2.1Ghz, 28 core/socket, 2 sockets, 2 threads per core	
		DRAM	DDR4 384GB total = 12 DIMMs 32GB@2666Mhz	
		Network	10Gbit on board Intel NIC	
	Software	OS	Fedora 27	
		Kernel	Kernel: 4.16.6-202.fc27.x86_64	
		JDK	Oracle Hotspot JDK (JDK1.8 u131)	
	Workload	Benchmark	Cassandra-Stress	
		Cassandra-Stress Instances	1	14
		Command line to write database	cassandra-stress user profile/root/cassandra_4.0/tools/cqlstress-insanity-example.yaml ops\(insert=1\) n=1250000000 cl=ONE no-warmup -pop seq=1..1250000000 -mode native cql3 -node <ip_addr> -rate threads=10	cassandra-stress user profile/root/cassandra_4.0/tools/cqlstress-insanity-example.yaml ops\(insert=1\) n=100000 cl=ONE no-warmup -pop seq=1..100000 -mode native cql3 -node <ip_addr> -rate threads=10
		Command line to read database	cassandra-stress user profile=/root/cassandra_4.0/tools/cqlstress-insanity-example.yaml ops\(simple1=1\) duration=10m cl=ONE no-warmup -pop dist=UNIFORM\(1.. 1250000000\) -mode native cql3 –node <ip_addr> -rate threads=300	cassandra-stress user profile=/root/cassandra_4.0/tools/cqlstress-insanity-example.yaml ops\(simple1=1\) duration=3m cl=ONE no-warmup -pop dist=UNIFORM\(1..100000\) -mode native cql3 –node <ip_addr> -rate threads=320


intel